

pi_ager install on Raspberry Pi 3/Pi 4/ Pi zero w under Pi OS (32-bit) with desktop and recommended software or Pi OS Lite

- **For Pi 4/3:** Download and install Raspberry Pi OS with desktop and recommended software from <https://www.raspberrypi.org/software/operating-systems/>
- **For Pi zero:** Download and install Raspberry Pi OS Lite from <https://www.raspberrypi.org/software/operating-systems/>

- Enable SSH for remote access

sudo touch /boot/ssh

- Setup WLAN configuration

Generate file wpa_supplicant.conf in /boot:

```
country=DE
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="WLAN SSID"
    psk="WLAN PASSWORT"
}
```

- Edit config.txt in /boot to support I2C and SPI devices:
Additional overlays and parameters are documented /boot/overlays/README
Use Pi-Ager Pins 11/13 GPIO 17/27 for I2C
dtoverlay=i2c-gpio,bus=3,i2c_gpio_sda=17,i2c_gpio_scl=27
Use Pi-Ager Pin 16 for MCP3204
dtoverlay=spi1-1cs,cs0_pin=16
- Add in /boot/cmdline.txt at the end of line this to enable USB camera with fswebcam :
dwc_otg.fiq_fsm_mask=0x3
- Reboot system
- Edit /etc/modules to load i2c-dev at boot, add this line :
i2c-dev
- Add file :
sudo touch /etc/modprobe.d/raspi-blacklist.conf
- Get a copy from Pi-Ager repository to your local system:
git clone -b entwicklung <https://github.com/Tronje-the-Falconer/Pi-Ager>
All project file are now in the folder ./Pi-Ager/
- Copy setup.txt from local repository to /boot/ and edit it as needed.
- Copy /etc/modprobe.d/Pi-Ager_i2c_off.conf.on from local repository to /etc/modprobe.d/

- Reboot system

- Install lighttpd:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install lighttpd
sudo systemctl status lighttpd
```

```
sudo nano /etc/lighttpd/lighttpd.conf
and change Parameter
```

```
server.document-root = "/var/www/html"
to
server.document-root = "/var/www"
```

```
sudo usermod -G www-data -a pi
sudo chown -R www-data:www-data /var/www
sudo chmod -R 755 /var/www
```

- Reboot system

For testing the web server, generate html-page:

```
sudo nano /var/www/test.html
with content:
```

```
<html>
<head><title>Test-Seite</title></head>
<body>
<h1>Das ist eine Test-Seite.</h1>
</body>
</html>
```

Enter your IP Address (or localhost) into the browser followed by /test.html

In addition we need .htcredentials to contain user and password.

For that we use the Online-Tool <https://websistent.com/tools/htdigest-generator-tool/>

Username: pi-ager

REALM: Pi-Ager

Password: raspberry

Caution! All entries are case sensitive!

Open this file now

```
sudo nano /var/.htcredentials
```

and fill in the string output from the generator tool.

Save file with "STRG+o", "RETURN" and close with "STRG+x"

Now we have to setup the password authentication in lighttpd:

```
sudo nano /etc/lighttpd/conf-available/05-auth.conf
```

The following lines are added under `server.modules += („mod_auth“)` :

```
auth.backend          = "htdigest"
auth.backend.htdigest.userfile = "/var/.htcredentials"

auth.require          = ( "/settings.php" =>
    (
        "method" => "digest",
        "realm" => "Pi-Ager",
        "require" => "user=pi-ager"
    ),
    "/admin.php" =>
    (
        "method" => "digest",
        "realm" => "Pi-Ager",
        "require" => "valid-user"
    ),
    "/webcam.php" =>
    (
        "method" => "digest",
        "realm" => "Pi-Ager",
        "require" => "valid-user"
    ),
    "/notification.php" =>
    (
        "method" => "digest",
        "realm" => "Pi-Ager",
        "require" => "valid-user"
    )
)
```

Then we activate this modul:

```
sudo lighty-enable-mod auth
```

In addition we have to edit :

```
sudo nano /etc/lighttpd/conf-available/15-fastcgi-php.conf
```

add at the end of the line

```
"broken-scriptfilename" => "enable"
```

a “,” and in a new line

```
"allow-x-send-file" => "enable"
```

Save end exit nano.

Now enable these modules:

```
sudo lighty-enable-mod fastcgi
```

```
sudo lighty-enable-mod fastcgi-php
```

Now reload the the webserver:

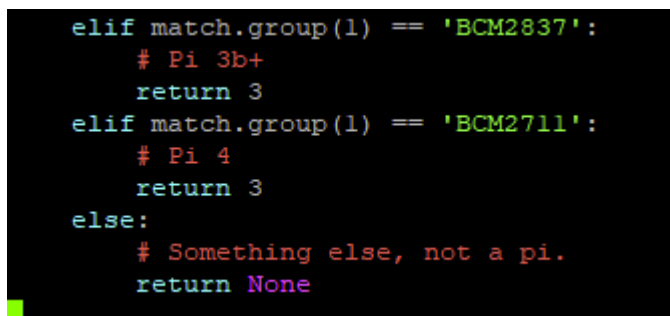
```
sudo service lighttpd force-reload
```

Now continue to install additional modules:

- Install Git
sudo apt install git
- Install pushover
cd ~
git clone <https://github.com/thibauth/python-pushover.git>
cd python-pushover
sudo pip3 install .
- Install smbus
sudo apt-get install python3-smbus
- Install sqlite3:
sudo apt-get install sqlite3
- Install DHT sensor support
sudo pip3 install Adafruit-DHT
- Install SHT1x sensors
sudo pip3 install pi-sht1x
- Install fswebcam:
sudo apt-get install fswebcam
- Install influxdb
sudo pip3 install influxdb
- Install php 7.3
sudo apt-get install php7.3-common php7.3-cgi php7.3 php7.3-sqlite3
- Install additional modules for php7.3:
sudo apt-get install php7.3-apcu php7.3-fpm php7.3-mbstring php7.3-phpdebug
- Install wiringpi:
sudo apt-get install wiringpi
- Install wiringpi new version with Pi4 support :
cd /tmp
wget <https://project-downloads.drogon.net/wiringpi-latest.deb>
sudo dpkg -i wiringpi-latest.deb
- Copy gpio to /usr/local/bin
sudo cp /usr/bin/gpio /usr/local/bin
sudo chmod 4755 /usr/local/bin/gpio
- Install PiShrink
wget <https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh>
chmod +x pishrink.sh
sudo mv pishrink.sh /usr/local/bin

- Nextion serial client (HMI Display support)
`sudo pip3 install nextion`
- php zip support:
`sudo apt-get update`
`sudo apt-get install php-zip`
- Install lsof command:
`sudo apt update`
`sudo apt install lsof`
- Install Locale en-GB and de-DE UTF-8 using
`sudo raspi-config`
- Enable Serial Interface, disable login, needed for HMI Nextion Display
`sudo raspi-config`
- Install zip and unzip:
`sudo apt install zip unzip`
- Workaround for Adafruit_DHT for Pi4:
In `"/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/platform_detect.py"`, you can add the followings at line #112 in the elif ladder, so it should workaround the issue.

```
elif match.group(1) == 'BCM2711':
    return 3
```



```
elif match.group(1) == 'BCM2837':
    # Pi 3b+
    return 3
elif match.group(1) == 'BCM2711':
    # Pi 4
    return 3
else:
    # Something else, not a pi.
    return None
```

- Unblock wifi for Pi4, add rfkill unblock wifi:
`cd /etc`
`sudo nano rc.local`

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
rfkill unblock wifi

exit 0
```

- Generate/edit crontab to prepare for automatic enable pi-ager_backup.sh

```
pi@pi-ager:~$ sudo crontab -l
#-----
# Shell variable for cron
SHELL=/bin/bash
# PATH variable for cron
PATH=/usr/local/bin:/usr/local/sbin:/sbin:/usr/sbin:/bin:/usr/bin:/usr/bin/X11
#M S T M W Befehl
#-----
#* */6 * * * /usr/bin/flock -w 0 /var/run/pi-ager_backup.pid /usr/local/bin/pi-ager_backup.sh >> /var/log/pi-ager_backup
.log 2>&1
#*/2 * * * * logger "greetings from the crontab" > /dev/null 2>&1
```

Use visudo to edit /etc/sudoers, so that the www-data User (User of Website) can execute /var/sudowebscript.sh :

sudo visudo

and then in sudoers following

...

#User privilege specification

root ALL=(ALL:ALL) ALL

...

adding:

www-data ALL=NOPASSWD:/var/sudowebscript.sh

Save and exit.

- Now copy alle files and folders from your local git repository /var/www to /var/www/
- from local repository /opt/pi-ager/ to /opt/pi-ager/
- from local repository /var/sudowebscript.sh to /var/
- **sudo chown -R www-data:www-data /var/www**
- **sudo chown root:root /var/www/**
- **sudo usermod -G gpio -a www-data**
- **sudo chmod 666 /var/www/logs/logfile.txt**
- **sudo chown -R root:root /var/www/logs**

- `sudo chmod 755 /var/www/logs/`
- `sudo chmod 664 /var/www/config/pi-ager.sqlite3`
- `sudo chown -R www-data:www-data /var/www/config/`
- `sudo chmod 555 /var/sudowebscript.sh`
- from local repository `/usr/local/bin/*.sh` copy all to `/usr/local/bin/`
(`pi-ager_backup.sh`, `pi-ager_image.sh`, `setup_pi-ager.sh`)
Set `+x` mode to the scripts :
`sudo chmod +x /usr/local/bin/*.sh`
- from local repository `/lib/systemd/system` copy the following files to
`/lib/systemd/system/` :
`pi-ager_main.service`
`setup_pi-ager.service`
- Enable `setup_pi-ager.service` to initialize system with data from `/boot/setup.txt` after next
reboot:
`sudo systemctl enable setup_pi-ager`
`sudo reboot`